Teacher.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Working_with_Text_Files
{
    class Teacher
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Class { get; set; }
        public string Section { get; set; }
    }
}
```

Manage_Teacher_Info.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Working_with_Text_Files
{
    class Manage_Teacher_Info
    {
        public bool AddTeacher(Teacher teacher, bool append)
        {
            try
            {
                string path = Environment.CurrentDirectory;
                FileInfo fileInfo = new FileInfo(path + "\\TeacherDetails.txt")
                using (StreamWriter streamWriter = new
              StreamWriter(fileInfo.FullName,append))
                {

              streamWriter.WriteLine($"{teacher.Id}\t{teacher.Name}\t{teacher.Class}\t
              {teacher.Section}");
                    streamWriter.Flush();
                }
                return true;

            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.ToString());
                return false;
            }


        }

        public bool UpdateTeacher(int id)
```

```
{
    List<Teacher> teacher = new List<Teacher>();
    try
    {
        string path = Environment.CurrentDirectory;
        FileInfo fileInfo = new FileInfo(path + "\\TeacherDetails.txt");
        string[] lines = File.ReadAllLines(fileInfo.FullName);

        foreach (var line in lines)
        {
            Teacher t1 = new Teacher();
            string[] values = line.Split('\t');
            t1.Id = Convert.ToInt32(values[0]);
            t1.Name = values[1];
            t1.Class = values[2];
            t1.Section = values[3];
            teacher.Add(t1);
        }
        if (teacher != null)
        {

            var dataToDelete = teacher.Where(x => x.Id ==id).FirstOrDefault();
            teacher.Remove(dataToDelete);
            fileInfo.Delete();
            Teacher t2 = new Teacher();
            Console.WriteLine("Enter updated details:");
            Console.Write("Enter Teacher Id:");
            t2.Id = Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter Teacher Name:");
            t2.Name = Console.ReadLine();
            Console.Write("Enter Teacher Class:");
            t2.Class = Console.ReadLine();
            Console.Write("Enter Teacher Section:");
            t2.Section = Console.ReadLine();
            teacher.Add(t2);
            foreach (var t in teacher)
            {
                AddTeacher(t, true);
            }
        }

        return true;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        return false;
    }

}

public void PrintAllTeacherDetails()
{
    try
    {

        string path = Environment.CurrentDirectory;
        FileInfo fileInfo = new FileInfo(path + "\\TeacherDetails.txt");

        string[] lines = File.ReadAllLines(fileInfo.FullName);
        foreach (var line in lines)
        {
```

```
                    Console.WriteLine(line);
                }
            }
            catch (Exception ex)
            {
              Console.WriteLine("No teacher Record found,Try adding Teacher
              Information first");
                Console.WriteLine(ex.Message);
            }
        }
    }
}
```

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Working_with_Text_Files
{
    class Program
    {
        static void Main(string[] args)
        {
        start: Console.Write("Enter 1.To add Teacher Details 2.To Update Teacher
Details 3.Display all Teacher Info:");
            int choice = Convert.ToInt32(Console.ReadLine());
            Manage_Teacher_Info manageTeacher = new Manage_Teacher_Info();
            switch (choice)
            {
                case 1:
                    Console.Write("Enter Teacher Id:");
                    int teach_id = Convert.ToInt32(Console.ReadLine());
                    Console.Write("Enter Teacher Name:");
                    string teach_name = Console.ReadLine();
                    Console.Write("Enter Teacher Class:");
                    string teach_class = Console.ReadLine();
                    Console.Write("Enter Teacher Section:");
                    string teach_section = Console.ReadLine();
                    var flag = manageTeacher.AddTeacher(new Teacher { Id = teach_id,
Name = teach_name, Class = teach_class,Section=teach_section }, true);
                    if (flag)
                    {
                        Console.WriteLine("Teacher Details Added Succesfully");
                    }
                    break;
                case 2:
                    Console.Write("Enter Teacher Id to be Updated:");
                    int id1 = Convert.ToInt32(Console.ReadLine());
                    var flag1 = manageTeacher.UpdateTeacher(id1);
                    if (flag1)
                    {
                        Console.WriteLine("Teacher Details Updated Successfully");
                    }
```

```
                    break;
                case 3:

                    manageTeacher.PrintAllTeacherDetails();

                    break;
                default:
                    Console.WriteLine("Invalid Choice");
                    break;
            }
            Console.WriteLine("Do You Want to Continue:(yes/no)?");
            string response = Console.ReadLine();
            if (response.ToLowerInvariant() == "yes")
            {
                goto start;
            }


        }
    }
}
```